OK TO ENTER: /AS/ (09/10/2008)

Amendments to the Claims

Please amend claim 1-29, as shown below.

1. (Canceled)

2. (Currently amended) A method for emulating a plurality of virtual timers in a virtual computer system operating on a physical computer, the physical computer having a timer ~~one or more timers~~ for keeping track of a real time for the physical computer, the virtual timers being programmable by guest software to generate a plurality of timer events, the method comprising:

receiving programming information from the guest software for programming a first virtual timer;

receiving programming information from the guest software for programming a second virtual timer;

determining when the first virtual timer is set to generate timer events according to the real time, based on the programming information received from the guest software;

determining when the second virtual timer is set to generate timer events according to the real time, based on the programming information received from the guest software;

wherein the generation of timer events falls behind the real time, so that a first plurality of timer events, including one or more timer events of the first virtual timer and one or more timer events of the second virtual timer, are set to have already occurred according to the real time, but the first plurality of timer events have not yet occurred in the virtual computer system; and

generating the first plurality of timer events in the same combined sequence as the timer events are set to occur according to the real time,

wherein a catch-up mode is used when the generation of timer events in the virtual computer system is behind the real time,

wherein a normal mode is used when the generation of timer events in the virtual computer system is caught up to the real time,

wherein, when the catch-up mode is used, the average rate of timer events in the virtual computer system exceeds the average rate at which timer events are set to occur according to the real time, ~~and~~

wherein, when the normal mode is used, the average rate of timer events in the virtual computer system is substantially the same as the average rate at which timer events are set to occur according to the real time, and

, wherein the real time being tracked by a timer in the physical computer.

3. (Currently amended) A method for emulating a plurality of virtual timers in a virtual computer system operating on a physical computer, the physical computer having a timer ~~one or more timers~~ for keeping track of a real time for the physical computer, the virtual timers being programmable by guest software to generate a plurality of timer events, the method comprising:

receiving programming information from the guest software for programming a first virtual timer;

receiving programming information from the guest software for programming a second virtual timer;

determining when the first virtual timer is set to generate timer events according to the real time, based on the programming information received from the guest software;

determining when the second virtual timer is set to generate timer events according to the real time, based on the programming information received from the guest software; and

generating timer events for the first virtual timer and the second virtual timer in the same combined sequence as the timer events are set to occur according to the real time,

wherein a catch-up mode is used when the generation of timer events in the virtual computer system is behind the real time,

wherein a normal mode is used when the generation of timer events in the virtual computer system is caught up to the real time,

wherein, when the catch-up mode is used, the average rate of timer events in the virtual computer system exceeds the average rate at which timer events are set to occur according to the real time, ~~and~~

wherein, when the normal mode is used, the average rate of timer events in the virtual computer system is substantially the same as the average rate at which timer events are set to occur according to the real time, and

, wherein the real time being tracked by a timer in the physical computer.

4. (Previously Presented) The method of claim 3 , wherein, when the normal mode is used, the interval between successive timer events in the virtual computer system is substantially the same as the interval between the same successive timer events as set according to the real time.

5. (Canceled)

6. (Previously Presented) The method of claim 3 , wherein the catch-up mode is entered substantially immediately when the generation of timer events in the virtual computer system falls behind the real time and the normal mode is entered substantially immediately when the generation of timer events in the virtual computer system catches up to the real time.

7. (Previously Presented) The method of claim 3, wherein, if the guest software attempts to read a read count value from a virtual timer, a returned count value is returned to the guest software that represents a returned time value that occurs after a preceding time value that is represented by a most recent preceding timer event and before a next time value that is represented by a next timer event to occur.

8. (Previously Presented) The method of claim 7, wherein the returned time value falls proportionately between the preceding time value and the next time value, based on the proportion at which the real time of the attempted reading of the read count value falls between

the real time at which the most recent preceding timer event was generated and the real time at which the next timer event is scheduled to be generated.

9. (Previously Presented) The method of claim 3 , wherein the method is performed by keeping track of an apparent time, which represents the time as it would appear to the guest software, as indicated by the virtual timers.

10. (Previously Presented) The method of claim 3, wherein the method is performed using a timer event queue.

11 - 15. (Canceled)

16. (Currently amended) A computer program embodied in a computer-readable storage medium, the computer program being executable on a physical computer as part of a virtual computer system, the physical computer having a timer ~~one or more timers~~ for keeping track of a real time for the physical computer, the virtual computer system comprising one or more timer emulators for emulating a plurality of virtual timers, each of the plurality of virtual timers generating one or more timer events, the computer program comprising:

a time coordinator for coordinating the respective timer events of the plurality of virtual timers, the time coordinator:

determining how each of the plurality of virtual timers has been programmed;

based on how each of the virtual timers has been programmed, determining a relative sequence of timer events as set according to the real time; and

notifying the one or more timer emulators when each of the plurality of virtual timers is to generate a timer event, so that the timer events are generated in the same combined sequence as the timer events are set to occur according to the real time,

wherein the time coordinator has a catch-up mode that is used when the generation of timer events in the virtual computer system is behind the real time, and a normal mode that is used when the generation of timer events in the virtual computer system is caught up to the real time, wherein, when the time coordinator is in the catch-up mode, the average rate of timer events in the virtual computer system exceeds the average rate at which timer events are set to occur according to the real time, ~~and~~

wherein, when the time coordinator is in the normal mode, the average rate of timer events in the virtual computer system is substantially the same as the average rate at which timer events are set to occur according to the real time, and

, wherein the real time being tracked by a timer in the physical computer.

17. (Previously Presented) The computer program of claim 16, wherein, when the time coordinator is in the catch-up mode, the interval between successive timer events in the virtual computer system is substantially proportional to the interval between the same successive timer events as set according to the real time.

18. (Previously Presented) The computer program of claim 16, wherein, when the time coordinator is in the normal mode, the interval between successive timer events in the virtual computer system is substantially the same as the interval between the same successive timer events as set according to the real time.

19. (Canceled)

20. (Previously Presented) The computer program of claim 16, wherein the time coordinator enters the catch-up mode substantially immediately when the generation of timer events in the virtual computer system falls behind the real time and the time coordinator enters the normal mode substantially immediately when the generation of timer events in the virtual computer system catches up to the real time.

21. (Previously Presented) The computer program of claim 16 , wherein, if a software entity attempts to read a read count value from a virtual timer, the time coordinator provides a value to one of the timer emulators, which causes the timer emulator to return a returned count value to the software entity that represents a returned time value that occurs after a preceding time value that is represented by a most recent preceding timer event and before a next time value that is represented by a next timer event to occur.

22. (Previously Presented) The computer program of claim 21, wherein the returned time value falls proportionately between the preceding time value and the next time value, based on the proportion at which the real time of the attempted reading of the count value falls between the real time at which the most recent preceding timer event was generated and the real time at which the next timer event is scheduled to be generated.

23. (Canceled)

24. (Currently amended) A method for coordinating a plurality of virtual timers in a virtual computer system, the virtual computer system operating within a physical computer system, the physical computer system having a timer ~~one or more timers~~ for keeping track of a real time for the physical computer system, the method comprising:

receiving programming information for each of the virtual timers, indicating when each of the virtual timers is to generate timer events;

determining when each of the virtual timers is set to generate timer events according to the real time;

causing the virtual timers to generate timer events in the same combined sequence as the timer events are set to occur according to the real time;

determining an apparent time that appears to exist within the virtual computer system based on timing information provided by the virtual timers;

when the apparent time is substantially the same as the real time, generating timer events at substantially the same real time as the timer events are set to occur according to the real time; ~~and~~

when the apparent time is ~~substantially~~ behind the real time, generating timer events at a faster rate than the timer events are set to occur according to the real time, until the apparent time catches up to the real time, and

, wherein the real time being tracked by a timer in the physical computer.

25. (Currently Amended) The method of claim 24, wherein, when the apparent time is ~~substantially~~ behind the real time, the interval between successive timer events in the virtual computer system is substantially proportional to the interval between the same successive timer events as set according to the real time.

26. (Canceled)

27. (Previously Presented) The method of claim 24, wherein timer events are generated at a faster rate than the timer events are set to occur according to the real time substantially immediately when the apparent time falls behind the real time, and wherein timer events are generated at substantially the same real time as the timer events are set to occur according to the real time substantially immediately when the apparent time catches up to the real time.

28. (Previously Presented) The method of claim 24 , wherein, if a software entity within the virtual computer system attempts to read a read count value from a virtual timer, a returned count value is returned to the software entity that represents a returned time value that

occurs after a preceding time value that is represented by a most recent preceding timer event and before a next time value that is represented by a next timer event to occur.

29. (Previously Presented) The method of claim 28, wherein the returned time value falls proportionately between the preceding time value and the next time value , based on the proportion at which the real time of the attempted reading of the read count value falls between the real time at which the most recent preceding timer event was generated and the real time at which the next timer event is scheduled to be generated.